# CROSS-PROGRAM WEAPONS SYSTEM SOFTWARE ACQUISITION CAN SAVE BILLIONS
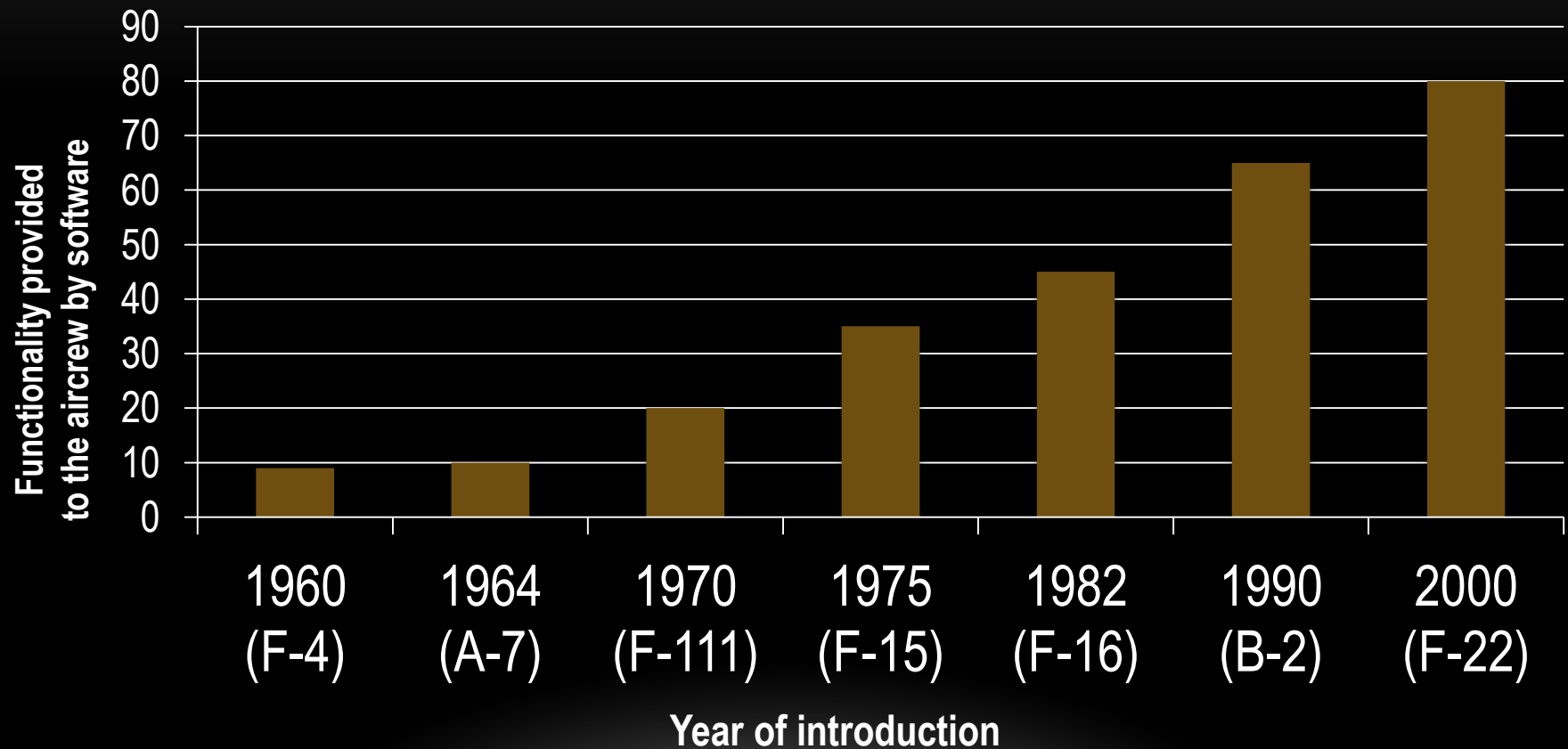
Rick Brennan

Operational Systems, Inc

May 17, 2012

| 1. REPORT DATE **17 MAY 2012** | 2. REPORT TYPE | 3. DATES COVERED **00-00-2012 to 00-00-2012** |
|---|---|---|
| 4. TITLE AND SUBTITLE **Cross-Program Weapons System Software Acquisition Can Save Billions** | | 5a. CONTRACT NUMBER |
| | | 5b. GRANT NUMBER |
| | | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | | 5d. PROJECT NUMBER |
| | | 5e. TASK NUMBER |
| | | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Operational Systems, Inc,1231 Collier Road NW, Suite D,Atlanta,GA,30318-2313** | | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES
**Presented at the 9th Annual Acquisition Research Symposium, May 16 - 17, 2012, Monterey, CA. U.S. Government or Federal Rights License**

14. ABSTRACT

15. SUBJECT TERMS

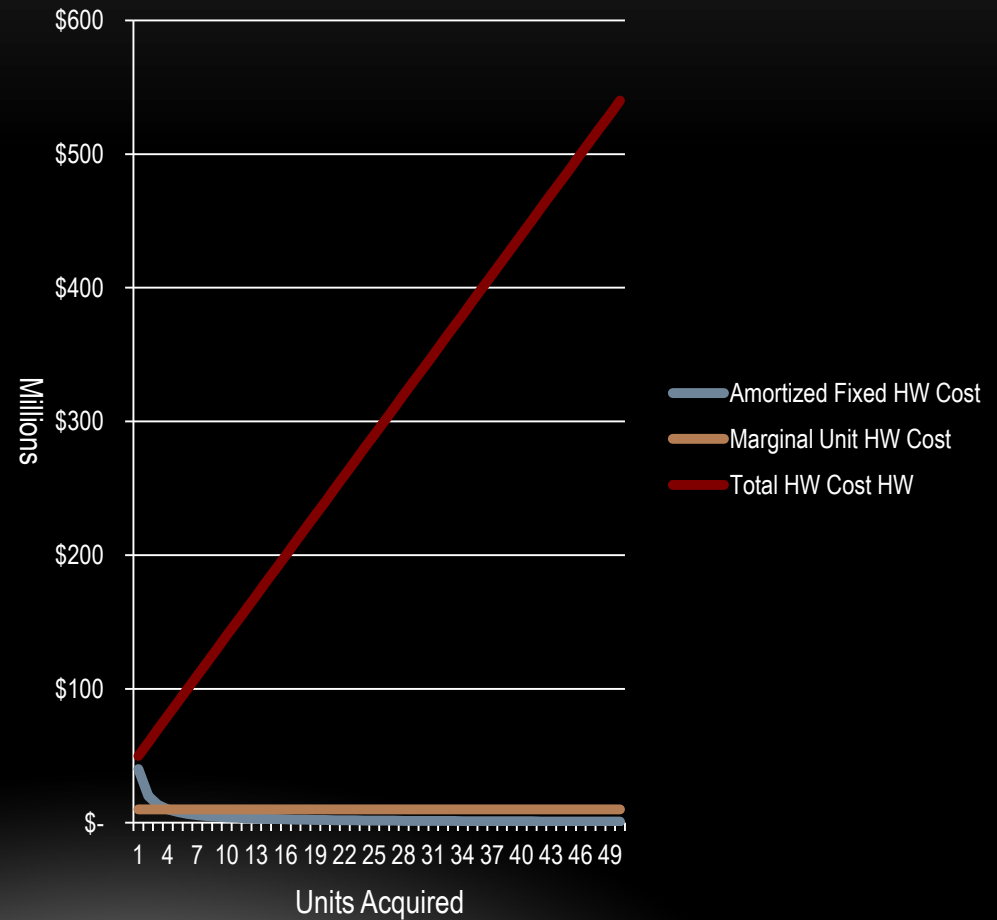| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT **unclassified** | b. ABSTRACT **unclassified** | c. THIS PAGE **unclassified** | **Same as Report (SAR)** | **10** | |

# SOFTWARE IN MILITARY AIRCRAFT



Source: NASA Study on Flight Software Complexity
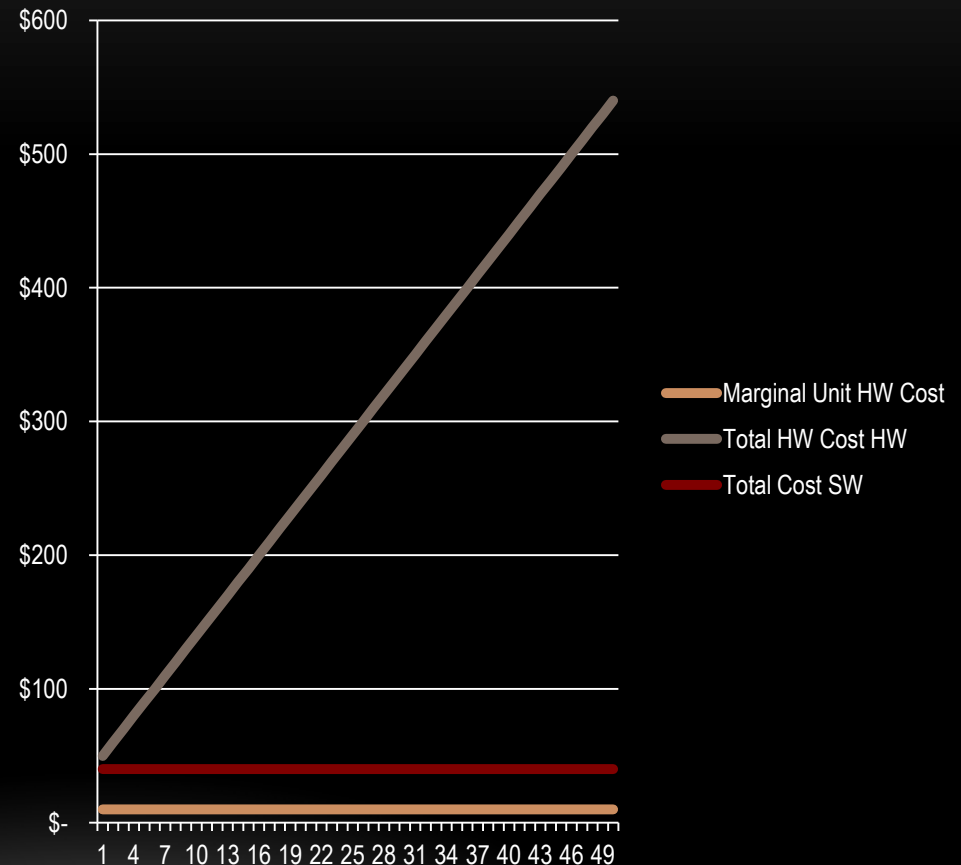
# Physical Goods Cost Behavior

- Notional example:

  - $40 million for development, integration, test, certification, and other non-recurring costs

  - $10 million for manufacturing, test, distribution, integration, and other recurring, per-unit costs

  - 1 core capability delivered

  - 50 units built
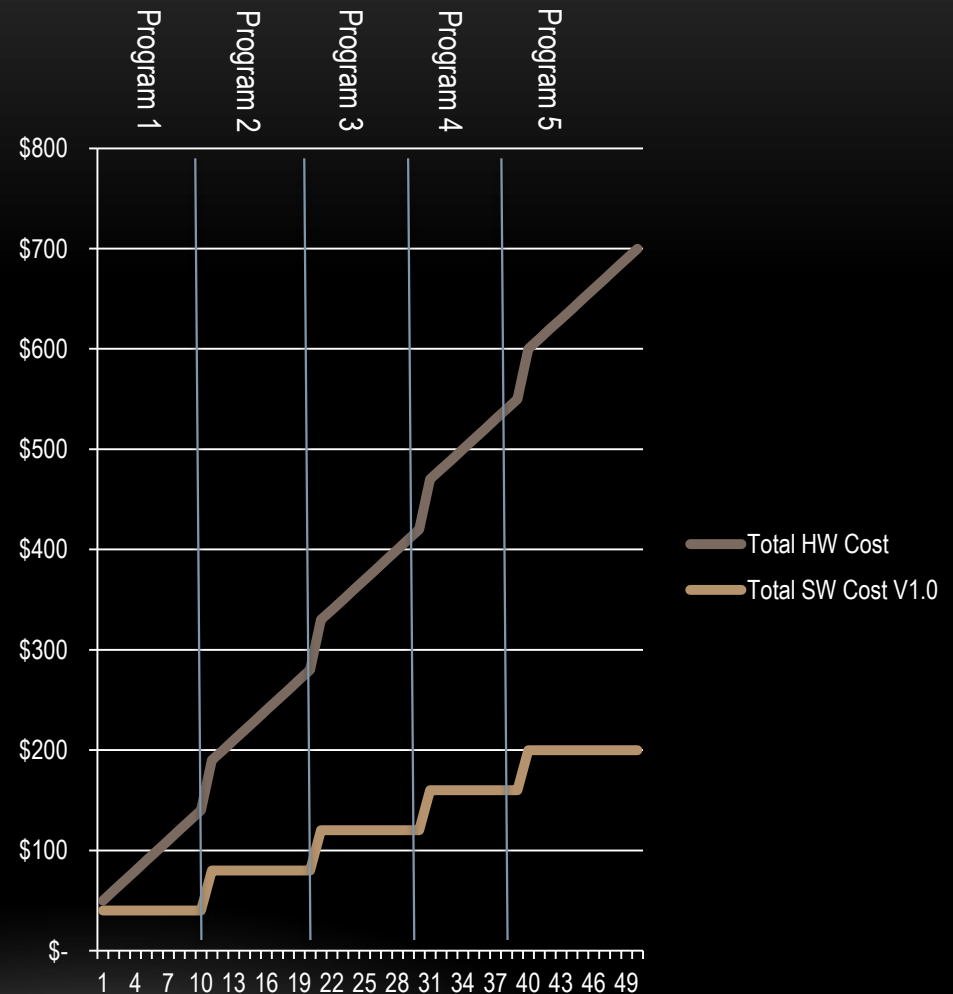
## Software Cost Behavior Comparison

Notional Example:

- $40 million for development, integration, test, certification, and other non-recurring costs

- $0 for manufacturing, test, distribution, integration, and other recurring, per-unit costs

- 1 core capability delivered

- 50 initial software instances built

# Economic Effect of Stovepiping

Notional Example:

- Each program builds a unique hardware and software product

- $40 million fixed development cost for each hardware product

- $10 million marginal manufacturing cost per hardware unit

- $40 million fixed development cost for each software product

- $5 million fixed development cost for each software version after V1.0

- $0 marginal manufacturing cost per software instance

Program 1  Program 2  Program 3  Program 4  Program 5

$800
$700
$600
$500
$400
$300
$200
$100
$-

1  4  7  10  13  16  19  22  25  28  31  34  37  40  43  46  49
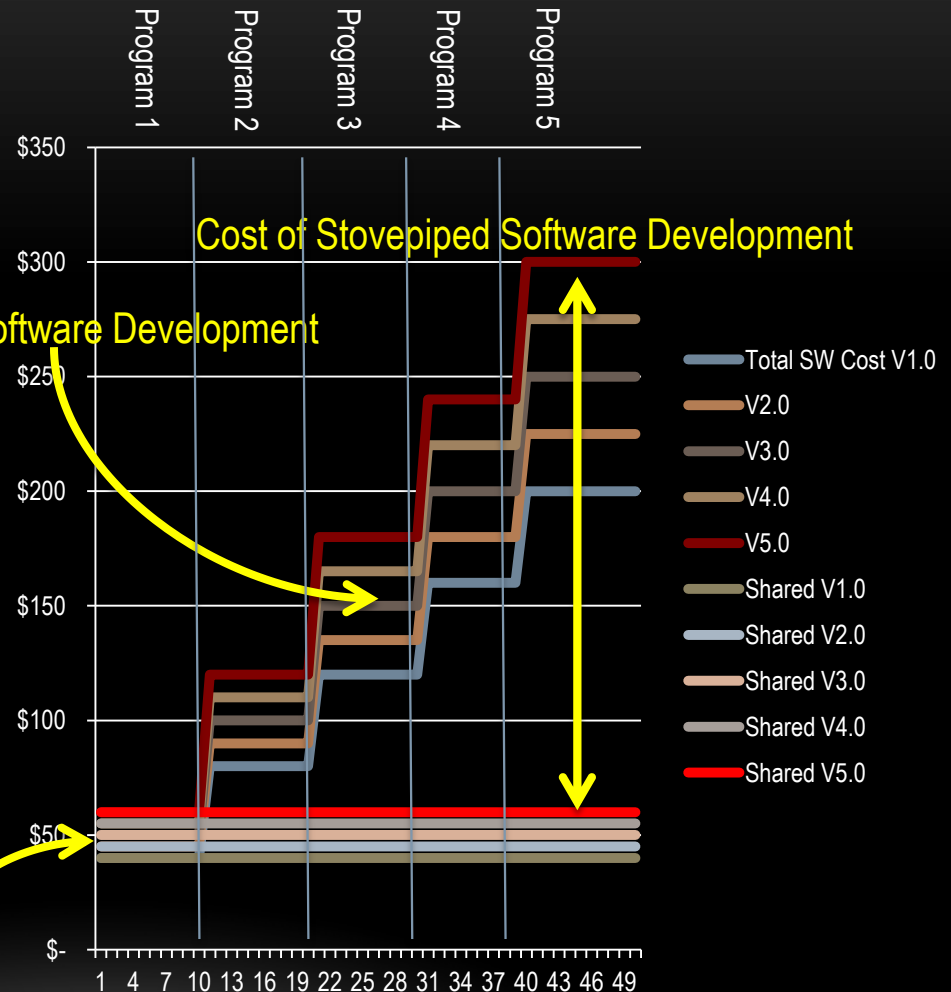
Total HW Cost
Total SW Cost V1.0

Stovepipe Economic Effects Compounded by Software Improvement Cycles

Notional Example:

- Each program builds a unique software product

- $40 million fixed development cost for each software product

- $5 million fixed development cost for each software version after V1.0

- $0 marginal manufacturing cost per software instance

Shared Intellectual Property

Cost of Stovepiped Software Development

Stovepiped Software Development

Program 1
Program 2
Program 3
Program 4
Program 5

Total SW Cost V1.0
V2.0
V3.0
V4.0
V5.0
Shared V1.0
Shared V2.0
Shared V3.0
Shared V4.0
Shared V5.0

# THE POWER OF SHARED INTELLECTUAL PROPERTY

- In the best case, once fixed costs for development, test, certification, and verification are paid, the cost to scale software is close to zero

- *Someone* has to pay those up front fixed costs

**But *Everyone* Doesn't!**

# CURRENT BARRIERS TO SHARING IP ACROSS PROGRAM BOUNDARIES

- Requirements development and management structure

- Funding structure

- Cultural interpretation of risk

- Business model

- Shortage of Government Lead Systems Integrator basic knowledge and skills

- Systems Command structure

- Systems Command skillset gaps

# MINIMIZING SHARING COSTS

- Identify appropriate intellectual property sharing systems targets

- Facilitate cross-program coordinated software development

  - Cooperative Product Line applications development

  - Community requirements management for frameworks and tools

- Shared Resources

  - Shared open source software infrastructure

  - Shared, cloud-based development and test environment

- Policy optimization

  - Financial incentives for programs that cooperate and collaborate

  - Re-thinking of the concept of risk

  - Streamlined IA and safety certification

- Business Model update

  - Separate physical vehicle acquisition from software applications acquisition

# QUESTIONS?